

To mesh or not to mesh. That is the question...

Sergio R. Idelsohn *, Eugenio Oñate

*ICREA and International Center for Numerical Methods in Engineering, CIMNE, Universidad Politècnica de Catalunya,
Gran Capitan s/n, 08034 Barcelona, Spain*

Received 4 April 2005; received in revised form 28 October 2005; accepted 1 November 2005

Abstract

In the last decade a family of methods called meshless methods has been developed both for structural and fluid mechanics problems. After these ideas, a possible classification for numerical formulations may be to separate the methods that make use of a standard finite element mesh (such as those made of tetrahedra or hexahedra), from those that do not need a standard mesh, namely the meshless methods. For solving a partial differential equation by a numerical method, a possible alternative may be either to use a mesh method or a meshless method. This paper discusses this issue to show that this choice is not, in the large majorities of the cases, the right question.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Meshless methods; Particle methods; Lagrangian formulations; Finite elements; Fluid mechanics

1. Introduction

After the big success of the finite element method (FEM), started in the 1960's by pioneers as Argyris et al. [1], Fraeijs de Veubeke [2] and Zienkiewicz and Cheung [3], the so called meshless methods have been developed in the last decade, both for structural [4,5] and fluid mechanics problems [6–13]. These new methods use the idea of a polynomial interpolant that fits a number of points minimizing the distance between the interpolated function and the value of the unknown points. The meshless techniques were proposed first by Nayroles et al. [4] were soon extended to solve problems in structural mechanics by Belytschko et al. [5] and in fluid mechanics by Oñate et al. [6–8]. Some of the many meshless methods developed in recent years may be found in [4–6,14,16–20]. In a previous paper the authors presented a numerical solution using a Lagrangian formulation and a point collocation method called the finite point method (FPM) [6]. Lately, the meshless ideas were generalized to take into account the finite element type approximations in order to obtain the same computing time in mesh generation as in the evaluation of the meshless connectivity [9]. This method, called the meshless finite element method (MFEM), make use of special finite element shape functions but has all the advantages of the meshless methods concerning the computing time of the nodal connectivity. The MFEM uses the extended Delaunay tessellation [10] to build a mesh combining elements of different polygonal (or polyhedral in 3D) shapes in a computing time which is linear with the number of nodal points. The MFEM is, in fact, a particular case of the finite element method, with elements of a general polyhedral shape. The MFEM shape functions have been used in a recent paper together with a particle method to solve fluid mechanics problems [11–13].

At this point, several questions arise:

- What is in fact a meshless method?
- Why are so many people trying to use meshless methods?

* Corresponding author. Tel.: +34 93 4017786; fax: +34 93 4016517.

E-mail addresses: sergio@cimne.upc.edu (S.R. Idelsohn), onate@cimne.upc.edu (E. Oñate).

- Which are the difficulties to generate a mesh?
- Is it the MFEM a meshless method or not?
- Is better to solve a problem with or without mesh?
- Which are the meshless difficulties?
- Is a particle method the same as a meshless method?
- May a particle method use a mesh?

This paper starts with a discussion about the problems to generate a mesh, the problems to use a meshless method and finishes by solving some fluid mechanics problems using the particle finite element method [11]. It must be noted that fluid mechanics problems with a moving free-surface are particular applications where the need of a particle method is clearer. In any case, the discussion and conclusions of this paper are valuable for many other applications.

2. Generate a mesh or “generate a meshless”

2.1. Why a meshless method?

In order to start the discussion it is interesting to ask ourselves: Why are we trying to use meshless methods? Which are the difficulties to generate a mesh? Why are we looking for a new formulation? In many physical problems the geometry of the domain changes with time. This is the case, for instance, in fluid mechanics problems with free-surfaces where the domain is an unknown and changes in time, can break in many free-surfaces or many free-surfaces can be joined in one. This kind of problem needs a continuous update of the node connectivities because two points, which are close to each other in a time step, may be very far from each other in the next step. The continuous regeneration of a mesh is a necessity in this problem; hence, the mesh must be built up quickly and in a fixed and bounded number of operations. Fig. 1 summarizes a problem of this kind where a meshless method can be useful. Certainly, many other problems may be found in which a permanent update of the nodal connectivity is necessary.

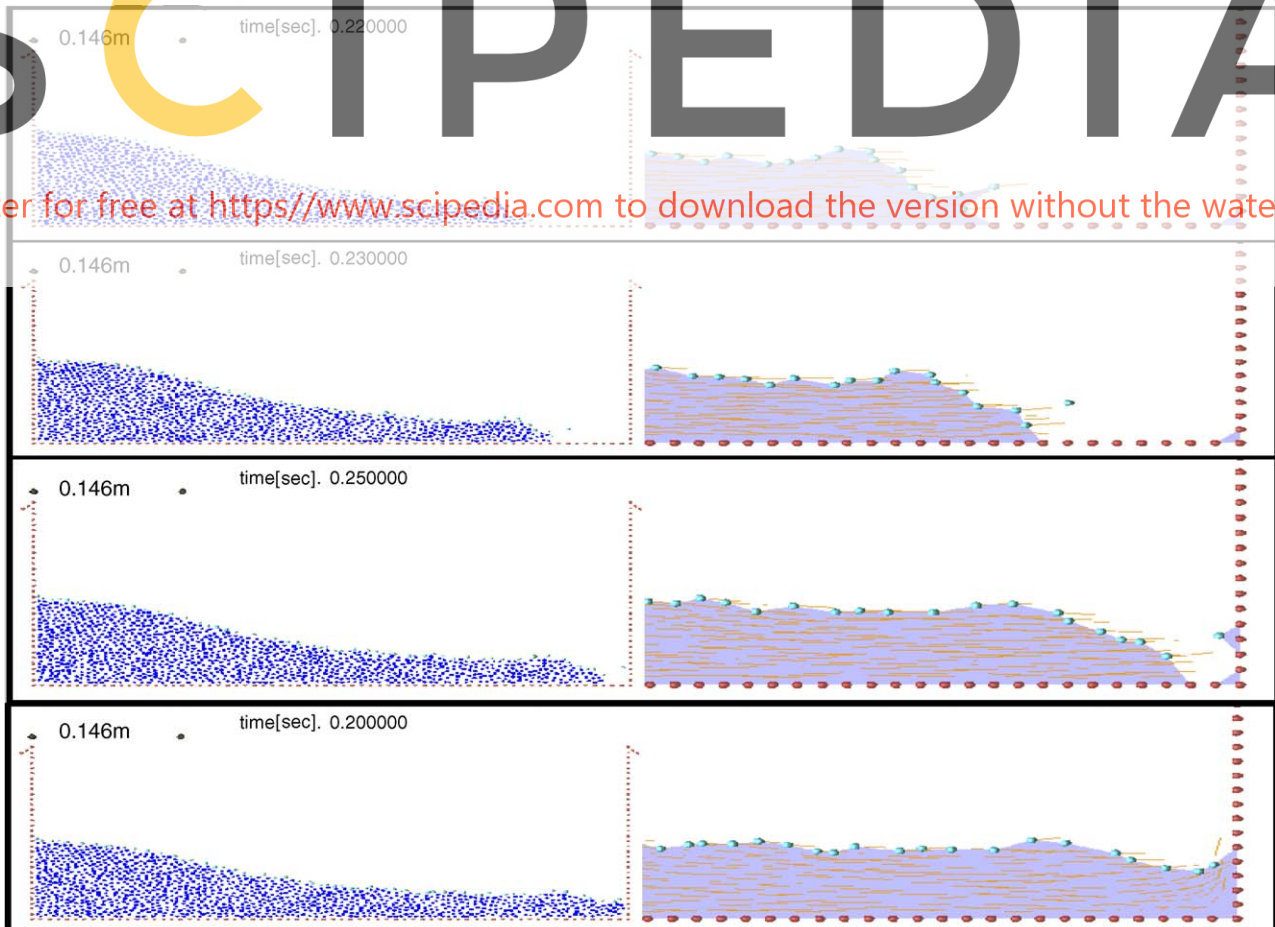


Fig. 1. Dam breaking: the solution of this problem is difficult with standard mesh-based method.

2.2. Some difficulties in build a mesh

In order to separate the problems, let us assume that we have a point distribution in space, and that we need to build a mesh with these points. Fig. 2 summarizes the main difficulties to build a mesh.

Mesheres must be conforming, the elements must have a reasonable good shape and the boundary contours must be respected.

In order to satisfy these three criteria, the existing mesh generators typically need large computing times and, in many cases, much human supervision is required to achieve a good result. This explain why for special problems, such as fluid flows with free-surfaces, alternative formulations solutions must be found.

2.3. Meshless difficulties

Meshless methods do not typically need a conforming mesh, but just the connectivities between neighbor nodes in order to build the approximation functions. Fig. 3 shows a classical way to build the nodal connectivities: all the nodes should lie within a sphere of radius r .

The main difficulties to build the nodal connectivity in meshless methods are precisely the advantages of the methods that use a mesh. These are the need for a variable distance between nodes, named h . Some times the variation of h is only in one direction (directional h). The problem of correctly fitting the boundary contours is a problem common to both methods. Some typical meshless connectivity problems are summarized in Fig. 3.

It must be noted that finding the node connectivities in meshless solutions as those presented in Fig. 3 may be as difficult as to solve the mesh generation problem and, in some cases, the computing time to generate the nodal connectivities in meshless problems is of the same order than the most difficult mesh generation problem. From this point of view, meshless methods are not as interesting as they initially were thought. Finding the meshless connectivities is a problem that in many

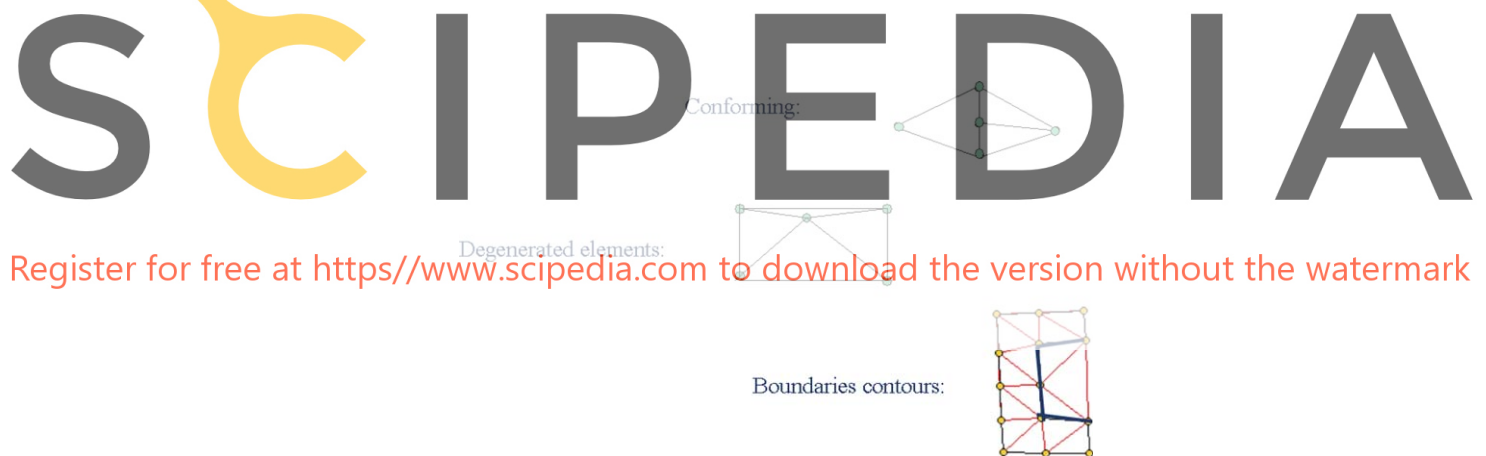


Fig. 2. Main difficulties to build a mesh.

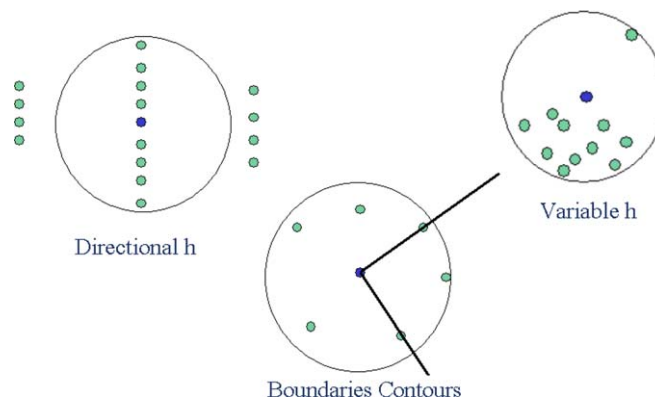


Fig. 3. Meshless connectivity problems.

Table 1
Mesh and meshless difficulty comparison

Problem	Standard mesh method	Meshless method
Conforming	Difficult	Simple
Degenerated elements	Difficult	Simple
Boundary contours	Difficult	Difficult
Directional h	Simple	Difficult
Variable h	Simple	Difficult

papers is not treated in its true dimension. As a result of this the use of a meshless method is useless in most cases and the same problem can be solved with a mesh more efficiently.

Table 1 summarizes those comparisons. Both, mesh and meshless methods have the boundary difficulty, and for variable point distributions, the advantages and disadvantages or both are “symmetrical”.

At this point it is useful to define what a meshless method is and other sometimes confusing entities in order to better understand the discussion.

2.4. Meshless methods

With this name are called a set of methods which main characteristic is defined by contraposition to the FEM, in the sense that they do not need a mesh in order to define the shape functions. An incomplete list of meshless methods is

- Non-structured finite differences [14],
- Diffusive elements [4],
- Element-free Galerkin [5],
- Smooth particle hydrodynamic [15],
- Reproducing kernel [16],
- Partition of unity [17],
- Finite point method [6–8],
- Meshless finite element [9],
- Finite sphere [18],
- Natural element [19],

SCIPEDIA

Register for free at <https://www.scipedia.com> to download the version without the watermark

Meshless methods are in fact something difficult to define. The condition of not existence of a mesh is rather diffuse. An acceptable definition is given in [9]:

A meshless method is an algorithm in which the definition of the shape functions depends only on the node positions.

This definition shows clearly that, for instance, the FEM is not a meshless method. Note that for a given node position, several triangulations are possible and then the shape functions may be different for the same node position.

Nevertheless, this definition is not enough to justify the use of a meshless method. In particular, we note that

A meshless method is useless without a fast evaluation of the nodal connectivity.

In a mathematical sense, fast means that the effort is bounded in time and linear in number of operations with the number of points in the domain. For this reason, in Ref. [11] a second statement was added to the previous one to define a useful meshless method.

A meshless method is an algorithm in which:

1. The definition of the shape functions depends only on the node positions.
2. The evaluation of the nodal connectivity is bounded in time and linear with the total number of nodes in the domain.

With these two statements, many of the self-called meshless methods are not truly meshless, or they need a correct definition of the nodal connectivity in order to be considered a meshless method.

On the other hand, meshless methods have shown some advantages in some particular applications, for instance: (a) they usually provide C^∞ interpolations which are convenient in numerical computations of fields or weak forms with higher order derivatives [25], (b) the non-local characters of meshfree interpolation has been proven to be effective in simulation of quasi-continua at nano-scale [26]. Nevertheless, in many problems the C^∞ interpolations or the non-local char-

acter of the interpolation do not bring in any specific advantages and they alone do not justify the use of a meshless methods.

2.5. A “good” numerical method

At this point it would be better to specify which are the non-standard features that “good” shape functions should have to be used in a numerical method:

1. The connectivity, which is necessary to evaluate the shape functions, must be evaluated in a bounded computing time of order n .
2. The shape function should depend only on the node positions.
3. The shape functions integrals and their derivatives must be evaluated in a simple manner (e.g., with a few Gauss points).
4. The shape functions must allow the representation of derivatives discontinuities.
5. Nodal interpolation. This means that the shape functions take a unit value at the nodes.

All of these five conditions may be found in both meshless or mesh based methods.

2.6. Particle methods

Another class of methods whose clarification is interesting at this stage is the so called particle methods.

A particle method is a method that represents the behavior of a physical problem by a collection of points named particles.

In a particle method each particle (Fig. 4) moves accordingly with its own mass and the internal/external forces applied to it. External forces are evaluated by the interaction with the neighboring particles by simple rules.

A particle may be a physical part of the domain (spheres, rocks, powder, etc.) or a specific part of the continuous domain previously defined. Note that particles may or may not coincide with the nodes in a mesh or with the points in a meshless method.

Another characteristic of particle methods is that all the physical and mathematical properties are attached to the particle itself and not to the elements as in the FEM. For instance physical properties such as viscosity or density, physical variables like velocity, temperature or pressure and also mathematical variables like gradients or volumetric deformations are assigned to each particle and the corresponding properties of the problem are obtained by averaging with the

Register for free at <https://www.scipedia.com> to download the version without the watermark

Particle methods are advantageous to treat discrete problems like granular materials but also to treat continuous problems in which there are possibilities of internal separations, contact problems or free-surfaces with breaking waves.

The best known particle method is probably the smooth particle hydrodynamics (SPH) method [15] in which the shape functions are generated using a linear kernel approximation. An enhanced SPH method can be derived by using the reproducing kernel particle method (RKPM) [16] in which the linear kernel functions are improved to reproduce high

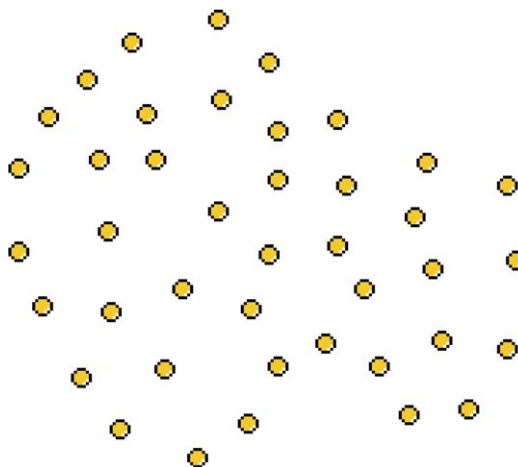


Fig. 4. Particle methods: all the information is defined by the particles.

polynomial orders. More recently, a new class of particle method called the particle finite element method (PFEM) [11–13] was presented by the authors. The PFEM combines ingredients of the standard particle methods with a background mesh where special finite element shape functions are used. The main features of the PFEM will be summarized in the following sections.

The most crucial characteristic of a particle method is probably that there is not a specified solution domain. The problem domain is defined by the particle positions and, hence, there is not a boundary surface or line. This is the reason why, when a differential equation is to be solved in order to evaluate the forces, the boundary surface needs to be identified in order to impose the boundary conditions (see Fig. 5).

Several algorithms may be used to define the boundary contours from the collection of particles. One of them is the alpha-shape algorithm [21] which will be used in the numerical test presented in this paper. In the alpha-shape algorithm:

All nodes defining an empty sphere with a radius $r(\mathbf{x})$ larger than $\alpha h(\mathbf{x})$ are considered as boundary nodes.

In this criterion, α is a parameter close to, but greater than one and $h(\mathbf{x})$ is the distance between two neighboring nodes (see Fig. 6).

One of the advantages of the alpha-shape idea is the easy way to determine when particles separate from the domain. This kind of phenomena may currently appear in fluid mechanics problems with free surfaces. Fig. 7 shows, for instance, the step before and after separation of a particle and a triangular surface from the main fluid domain.

It must be noted that in order to evaluate the internal forces between particles any classical approximation method may be used, including FEM, finite difference, meshless methods, etc. This means that a particle method may be used with or without a mesh, depending on the method chosen to evaluate the interacting forces.



Fig. 5. The domain is defined by the particles.

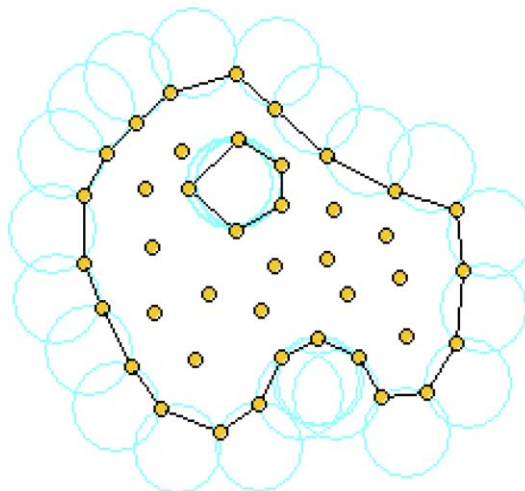


Fig. 6. Alpha-shape criterion to determine the boundary contours.

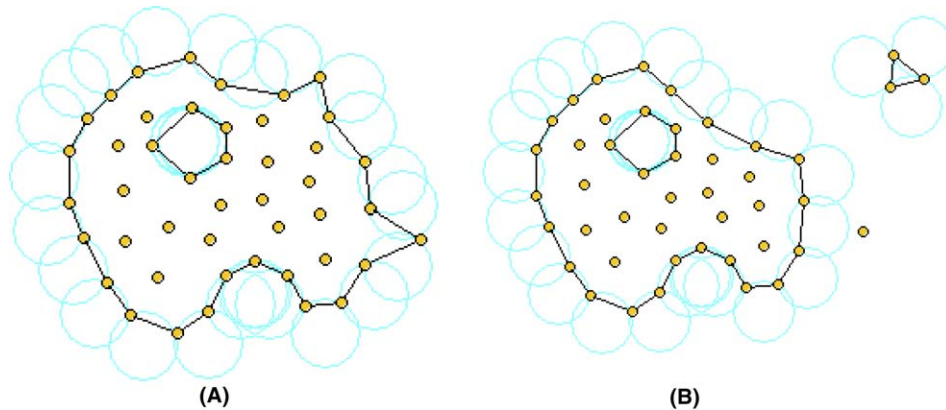


Fig. 7. Particle and domain separation detected by the alpha-shape algorithm: (A) before separation and (B) after separation.

3. The meshless finite element method (MFEM)

MFEM is the name given in Ref. [9] to a particular case of the finite element method with special shape functions. In the MFEM the domain is divided into spherical polyhedral elements. Non-sibsonian [22] shape functions are defined within each polyhedron. The main difference between the MFEM and the natural element method [19] is that in the latter the non-sibsonian shape functions are applied to all the domain, while in the MFEM they are applied only to each polyhedral element.

The name of meshless finite element method was introduced because the shape functions satisfy both main features of the FEM and the meshless methods: (a) the space is divided into elements with continuity of the shape functions but with discontinuity of the derivatives, and (b) the method satisfies the two statements of the meshless methods: the shape functions depends only on the node positions, and the evaluation of the nodal connectivity is bounded in time and is linear with the total number of nodes in the domain.

A full description of the MFEM may be found in [9,10]. A resume of the approach is presented next.

The meshless finite element method (MFEM) satisfies all the five statements previously defined for a good numerical method:

1. The nodal connectivities are evaluated in a bounded computing time of order $n^{1.5}$.
2. The shape functions depend only on the node positions.
3. The shape functions integrals and their derivatives are evaluated with a few Gauss points.
4. The shape functions allow the representation of derivatives discontinuities.
5. Nodal interpolation.

In the MFEM, the domain is divided in spherical polyhedral elements which have the advantages that the mesh is built very quickly using the extended Delaunay tessellation [10], independently of the node distribution, variable distance between nodes and without wrong elements (slivers). Fig. 8 shows some classical polyhedral elements and Fig. 9 the shape function definition and the corresponding function obtained for a pentagonal element.

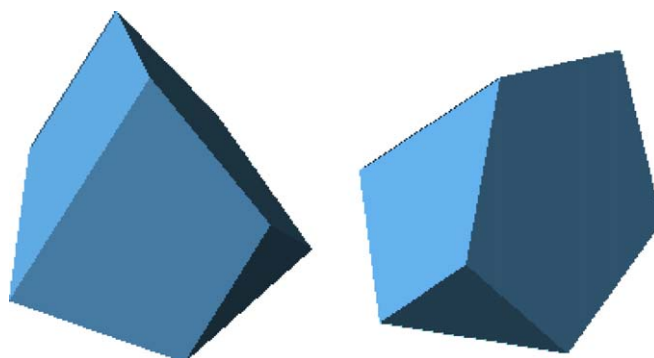


Fig. 8. Some particular case of spherical polyhedral elements used in the MFEM.

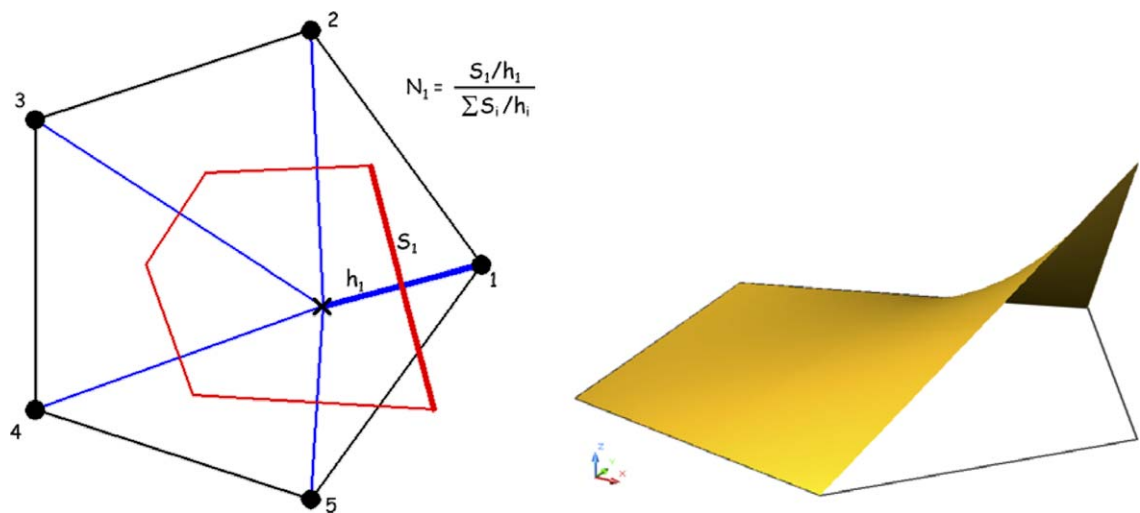


Fig. 9. Shape function definition and typical shape function for a pentagon element.

4. The particle finite element method (PFEM)

We name the particle finite element method to the method combining the following three features:

- 1. the information is particle-based, i.e., all the geometrical and mechanical information is attached to a particle,
- 2. the MFEM is used to compute the forces acting on each individual particle, and
- 3. the boundaries of the domain are defined via the alpha-shape method.

The combination of a particle method together with the MFEM and the alpha-shape method, avoids all the difficulties existing in both standard mesh and standard meshless methods (see Table 2).

Effectively, the PFEM may be seen as a particle method using special shape functions to evaluate the interacting forces between the particles. As a particle method, it does not have problems to define the boundary contours, because the contours are defined by the particles and they are not imposed by the problem itself. In other words, the boundary contours are defined by the order h distribution, instead of two neighbouring nodes. On the other hand, the shape functions inherit all the advantages of MFEM, this means: easy to solve problem with variable h distributions, directional h distributions and conforming meshes. In addition, meshes are generated with a speed of order n without degenerated elements (slivers).

The PFEM is therefore one of the few numerical methods that can avoid all the five difficulties to build the shape functions.

This conclusion refers to the main question stated at the beginning of this paper. We think that the correct question is not if it is better to build or not a mesh. The key issue is to have a method in which the evaluation of the nodal connectivity necessary to build good shape functions to solve numerically the problem is performed in order n . Otherwise, the method will be inefficient or limited only to homogeneous point distributions without change in the direction or the value of h .

5. The PFEM applied to solve incompressible fluid flows

In order to show the efficiency of the evaluation of the connectivity, PFEM will be used to deal with the incompressible Navier–Stokes equations. It must be noted that the PFEM may be used in many other applications but, it is in the fluid

Table 2
Comparison of mesh, meshless and PFEM difficulties

Problem	Standard mesh methods	Standard meshless methods	PFEM
Conforming	Difficult	Simple	Simple
Degenerated elements	Difficult	Simple	Simple
Boundary contours	Difficult	Difficult	Simple
Directional h	Simple	Difficult	Simple
Variable h	Simple	Difficult	Simple

flows, specially in presence of free surfaces, moving boundaries, breaking waves and fluid domain separations, where the advantages of the PFEM are shown more clearly.

The fluid will be assumed to be continuous and incompressible when is subjected to compression forces, but it has the possibility to separate under traction forces. This is the case of most physical fluids, such as water, oils and other fluids with low rate of surface tractions.

Both, fluid and solid boundaries will be modeled by an arbitrary number of particles. On each particle, the acting forces will be the gravity force and the interacting forces with the neighbor particles. These forces will be evaluated by solving the Navier–Stokes equations in a mesh. At each time step, a new mesh is generated in order to define the shape functions to solve the differential equations. This mesh is only useful for the definition of the interacting forces and it vanishes once the forces are evaluated. As previously stated, the interpolation functions chosen are the meshless finite element method shape functions. The boundary surface is defined using the alpha-shape method explained above. The evaluation of the interacting forces between particles is described next.

5.1. Particle position update

The particle positions are updated by solving the Lagrangian form of the Navier–Stokes equations.

Let X_i the initial position of a particle at time t^n .

Let x_i the final position of a particle at time t^{n+1} and the time increment $\Delta t = t^{n+1} - t^n$.

The final position of the particle can be approximated by

$$x_i = X_i + u_i^{n+1} \Delta t,$$

where $u_i(x, t^{n+1}) = u_i^{n+1}$ is the velocity of the particle at time t^{n+1} .

In the same way the displacement of the particle $d_i(x, t^{n+1}) = d_i^{n+1} = u_i^{n+1} \Delta t$.

5.2. Lagrangian governing equations in a viscous fluid flow

In the final x_i position, the mass and momentum conservation equations can be written in the Lagrangian form as

Mass conservation:

$$\frac{D\rho}{Dt} + \rho \frac{\partial u_i}{\partial x_i} = 0, \quad (1)$$

Momentum conservation:

$$\rho \frac{Du_i}{Dt} = -\frac{\partial}{\partial x_i} p + \frac{\partial}{\partial x_j} \tau_{ij} + \rho f_i, \quad (2)$$

where ρ is the density, u_i are the Cartesian components of the velocity field, p is the pressure, τ_{ij} is the deviator stress tensor, f_i is the source term (normally the gravity) and $\frac{D\phi}{Dt}$ represents the total or material time derivative of a function ϕ .

For Newtonian fluids the stress tensor τ_{ij} may be expressed as a function of the velocity field through the viscosity μ by

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_l}{\partial x_l} \delta_{ij} \right). \quad (3)$$

For near incompressible flows $\left(\frac{\partial u_i}{\partial x_i} \ll \frac{\partial u_k}{\partial x_l} \right)$ the momentum equations reduce to

$$\rho \frac{Du_i}{Dt} = -\frac{\partial}{\partial x_i} p + \frac{\partial}{\partial x_j} \tau_{ij} + \rho f_i \approx -\frac{\partial}{\partial x_i} p + \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \rho f_i. \quad (4)$$

5.3. Boundary conditions

On the boundaries, the standard boundary conditions for the Navier–Stokes equations are

$$\tau_{ij} v_j - p v_i = \bar{\sigma}_{ni} \quad \text{on } \Gamma_\sigma,$$

$$u_i v_i = \bar{u}_n \quad \text{on } \Gamma_n,$$

$$u_i \zeta_i = \bar{u}_t \quad \text{on } \Gamma_t,$$

where v_i and ζ_i are the components of the normal and tangent vector to the boundary.

5.4. Implicit–explicit time integration

Eq. (4) is integrated implicitly in time as

$$\rho \frac{Du_i}{Dt} \approx \rho \frac{u_i(x_i, t^{n+1}) - u_i(X_i, t^n)}{\Delta t} = \rho \frac{u_i^{n+1} - u_i^n}{\Delta t} = \left[-\frac{\partial}{\partial x_i} p + \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \rho f_i \right]^{n+\theta}, \quad (5)$$

where $[\phi(x, t)]^{n+\theta}$ means $\theta\phi(x, t^{n+1}) + (1-\theta)\phi(x, t^n) = \theta\phi^{n+1} + (1-\theta)\hat{\phi}^n$ and $\hat{\phi}^n = \phi(x, t^n)$ represents the value of the function at time t^n but at the final position x . For simplicity ϕ^n will be used instead of $\hat{\phi}^n$.

Only the case of $\theta = 1$ (full implicit) will be considered next. Other values as for instance $\theta = 1/2$ may be considered without major changes.

The time integrated equations become:

$$\rho \frac{u_i^{n+1} - u_i^n}{\Delta t} = \left[-\frac{\partial}{\partial x_i} p \right]^{n+1} + \left[\mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \rho f_i \right]^{n+1}. \quad (6)$$

The mass conservation is also integrated implicitly by

$$\frac{D\rho}{Dt} \approx \frac{\rho^{n+1} - \rho^n}{\Delta t} = -\rho^{n+1} \frac{\partial(u_i^{n+1})}{\partial x_i}. \quad (7)$$

5.5. The time splitting

The time integration of Eq. (6) presents some difficulties because it is a full coupled equation involving four degrees of freedom per node. When the fluid is incompressible or nearly incompressible advantages can be taken from the fact that in Eq. (7) the three components of the velocity are only coupled via the pressure. The fractional-step method will be used for the solution in time [23]. This basically consists in splitting each time step into two pseudo-times steps. In the first step, the implicit part of the pressure is avoided and this leads to decoupled equations in each of the velocity components. The implicit part of the pressure is added in a second step. The fractional-step algorithm for Eq. (7) is the following:

Split of the momentum equations.

$$\rho \frac{Du_i}{Dt} \approx \rho \frac{u_i^{n+1} - u_i^n}{\Delta t} = \rho \frac{u_i^{n+1} - u_i^* + u_i^* - u_i^n}{\Delta t} = \frac{1}{\rho} \frac{\partial}{\partial x_i} p^{n+1} + \frac{1}{\rho} \frac{\partial \tau_{ij}^{n+\theta}}{\partial x_j} + f_i, \quad (8)$$

where u_i^* are auxiliary variables termed fractional velocities defined by the split:

$$(A) \quad u_i^* = u_i^n + f_i \Delta t - \frac{\Delta t}{\rho} \frac{\partial}{\partial x_i} \gamma p^n + \frac{\Delta t}{\rho} \frac{\partial}{\partial x_j} \tau_{ij}^{n+\theta}, \quad (9)$$

$$(C) \quad u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho} \frac{\partial}{\partial x_i} (p^{n+1} - \gamma p^n), \quad (10)$$

where $p^n = p(x, t^n)$ is the value of the pressure at time t^n but evaluated at the final position and f_i is assumed to be constant in time.

In Eqs. (9) and (10) γ is a parameter giving the amount of pressure splitting, varying between 0 and 1. A larger value of γ means a small pressure split. In this paper γ will be fixed to 0 in order to have the larger pressure split and hence, a better pressure stabilization. Other values as, for instance $\gamma = 1$, may be used to derive high order schemes in time.

In the following we will choose $\gamma = 0$ and $\theta = 1$, then Eq. (9) becomes:

$$(A) \quad u_i^* - \frac{\Delta t}{\rho} \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i^*}{\partial x_j} \right) = u_i^n + f_i \Delta t. \quad (11)$$

The simplest way to introduce the incompressibility condition in a Lagrangian formulation is to write:

$$\rho^{n+1} = \rho^n = \rho^0 = \rho. \quad (12)$$

Then, the first term of Eq. (7) vanishes. Substituting Eq. (10) into Eq. (7) gives:

$$\frac{\rho}{\Delta t} \frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2} (p^{n+1}). \quad (13)$$

The three steps fractional method implemented in this work can be summarized by

$$\begin{aligned}
 \text{(A)} \quad & u_i^* - \frac{\Delta t}{\rho} \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i^*}{\partial x_j} \right) = u_i^n + f_i \Delta t \Rightarrow u_i^*, \\
 \text{(B)} \quad & \frac{\rho}{\Delta t} \frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2} (p^{n+1}) \Rightarrow p^{n+1}, \\
 \text{(C)} \quad & u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho} \frac{\partial}{\partial x_i} (p^{n+1}) \Rightarrow u_i^{n+1}.
 \end{aligned} \tag{14}$$

Further details on the fractional step method applied to the PFEM can be found in [11].

5.6. Spatial discretization via the meshless finite element method (MFEM)

The unknown functions are approximated using an equal order interpolation for all variables in the final configuration:

$$\begin{aligned}
 u_i &= \sum_l N_l(X, t) U_{il}, \\
 p &= \sum_l N_l(X, t) P_l.
 \end{aligned}$$

In matrix form:

$$\begin{aligned}
 u_i &= \mathbf{N}^T(X, t) \mathbf{U}_i \\
 p &= \mathbf{N}^T(X, t) \mathbf{P},
 \end{aligned}$$

or in compact form:

$$u_i = \mathbf{N}_i^T \mathbf{U} = \begin{bmatrix} \mathbf{N}^T & & \\ & \mathbf{N}^T & \\ & & \mathbf{N}^T \end{bmatrix} \mathbf{U}, \tag{15}$$

where \mathbf{N}^T are the MFEM shape functions and \mathbf{U} , \mathbf{P} the nodal values of the three components of the unknown velocity and the pressure, respectively.

It must be noted that the shape functions $\mathbf{N}(X, t)$ are functions of the particle coordinates. Then, the shape functions may change in time following the particles position. During a time step a mesh update may introduce changes in the shape function definition, which must be taken into account. During the time integration there are involving two times: t^n and t^{n+1} . The following notation will be used to distinguish between $\mathbf{N}(X, t^n)$ and $\mathbf{N}(X, t^{n+1})$:

$$\mathbf{N}(X, t^n) = \mathbf{N}^n \quad \text{and} \quad \mathbf{N}(X, t^{n+1}) = \mathbf{N}^{n+1}.$$

Nevertheless, the following hypothesis will be introduced: There is not mesh update at each time step. This means that if a mesh update is introduced at the beginning of a time step, the same mesh (but deformed) will be kept until the end of the time step. Mathematically this means:

$$\mathbf{N}(X, t^n) = \mathbf{N}(X, t^{n+1}).$$

The discretized equations are obtained by applying the Galerkin weighted residual method to Eq. (14) in the standard finite element fashion. Details of the discretized process can be found in [9].

5.7. Summary of a full iterative time step

A full time step may be described as follows: starting with the known values u^n and p^n for each particle, the computation of the new particle position involves the following steps:

1. Approximate u^{n+1} . (For the first iteration $u^{n+1} = 0$. For the subsequent iterations the value of u^{n+1} corresponding to the last iteration is taken.)
2. Move the particles to the x^{n+1} position and generate a mesh.
3. Evaluate the u^* velocity. It must be noted that the mass and stiffness matrices are separated in 3 blocks. Then, the momentum equations may be solved separately for u_x^* , u_y^* and u_z^* . For $\theta \neq 0$ (implicit) this involves the solution of 3 symmetric systems of equations. For $\theta = 0$ (explicit) the mass matrix may be lumped and inverted directly.
4. Evaluate the pressure p^{n+1} by solving a Laplacian equation.
5. Evaluate the velocity u^{n+1} . Go to 1 until convergence.

The Lagrangian split scheme described has some important advantages. The use of a Lagrangian formulation eliminates the standard convection terms present in Eulerian formulations. The convection terms are responsible for non-linearity, non-symmetry and non-self-adjoint operators, which require the introduction of high order stabilization terms to avoid numerical oscillations. All these problems are absent in this formulation. Only the nonlinearity remains due to the unknown of the final particle position. Another advantage is that the systems of equations to be solved are scalar (only one degree of freedom per node), symmetric and positive definite. Then, it is very easy to solve them using a symmetric iterative scheme (such as the conjugate gradient method). Finally, it must be noted that the use of an equal order approximation for both the velocity and the pressure terms introduce numerical instabilities in the pressure field that must be stabilized. A complete description of several stabilizing methods as well as the stabilization used in the examples presented in this paper may be found in [13,24].

5.8. Time stepping and iterative process

The time step length Δt was imposed to a variable value and evaluated at each time step.

The criterion to calculate the time step was the following: during the iterative implicit process, the time step may be as big as possible with the limitation that at the end of the iteration all elements have a positive volume. In this way, the mesh is preserved during the entire time step. This criterion is less restrictive than imposing a Courant number less than one.

In all the examples performed, a maximum of 3 iterations in the iterative process were need to reach a reasonable convergence.

6. Computing time with or without a mesh

The right question before deciding to solve a problem by a numerical method is: which is the computing time to solve a problem using either one or the other numerical method?

It is important to note that all the numerical methods, with or without a mesh, need the resolution of a system of equations which is the most computing time consuming. This system of equations is currently nonlinear and in transient problem must be solved at each time step.

Using as comparison the solution of an incompressible fluid flow problem, all the methods need at each time step to solve a nonlinear system of equations, which has (for the 3D mixed equal order formulation) 4 degree of freedom at each node. Being n the total number of nodes of the domains, this means a nonlinear system of $4n$ equations with $4n$ unknowns.

The fractional step method used in the PFEM formulation, reduces this problem to solve 4 nonlinear systems of n equations with n unknowns. The fractional step method is preferable to a monolithic one due to different reasons:

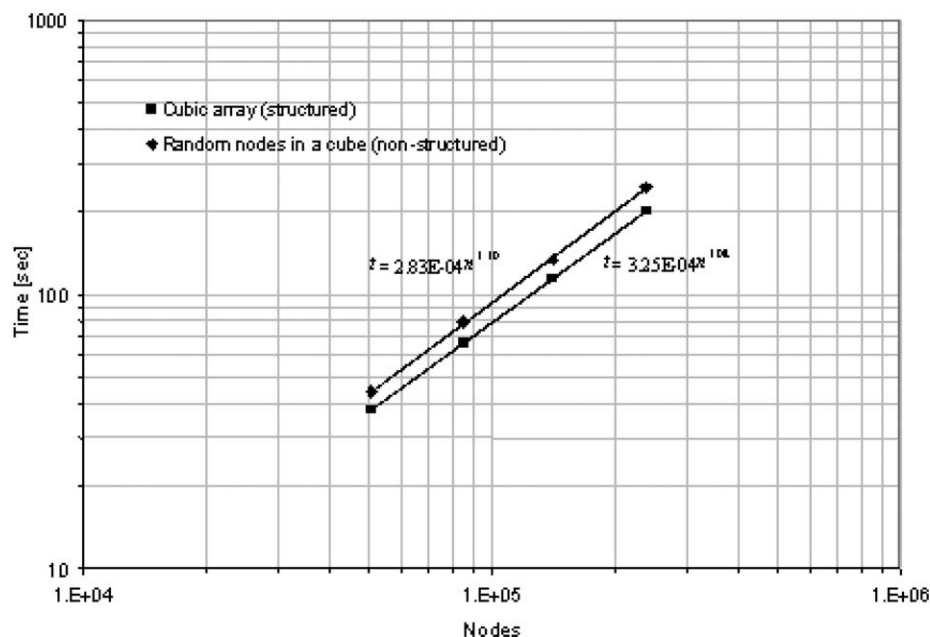


Fig. 10. Computing time in a standard PC for the mesh generation and the boundary recognition.

Table 3

Average computing time for a standard CFD method and the PFEM

n	Other methods without mesh update	PFEM	Increase (%)
10^4	30 s	101 s	236
10^5	20 min	35 min	75
10^6	13.3 h	16.1 h	21

1. The solution of any system of equation is nonlinear with the number of unknowns. Then is better to solve 4 ($n \times n$) systems than a $4n \times 4n$ one.
2. The four systems of equations in the fractional step method are better conditioned.
3. The four systems of equations in the fractional-step method are symmetrical and positive definite.

In the PFEM, the iteration computing time is of the same order than the standard fractional step method. The only difference is the evaluation of the EDT connectivity and the alpha-shape boundary recognition.

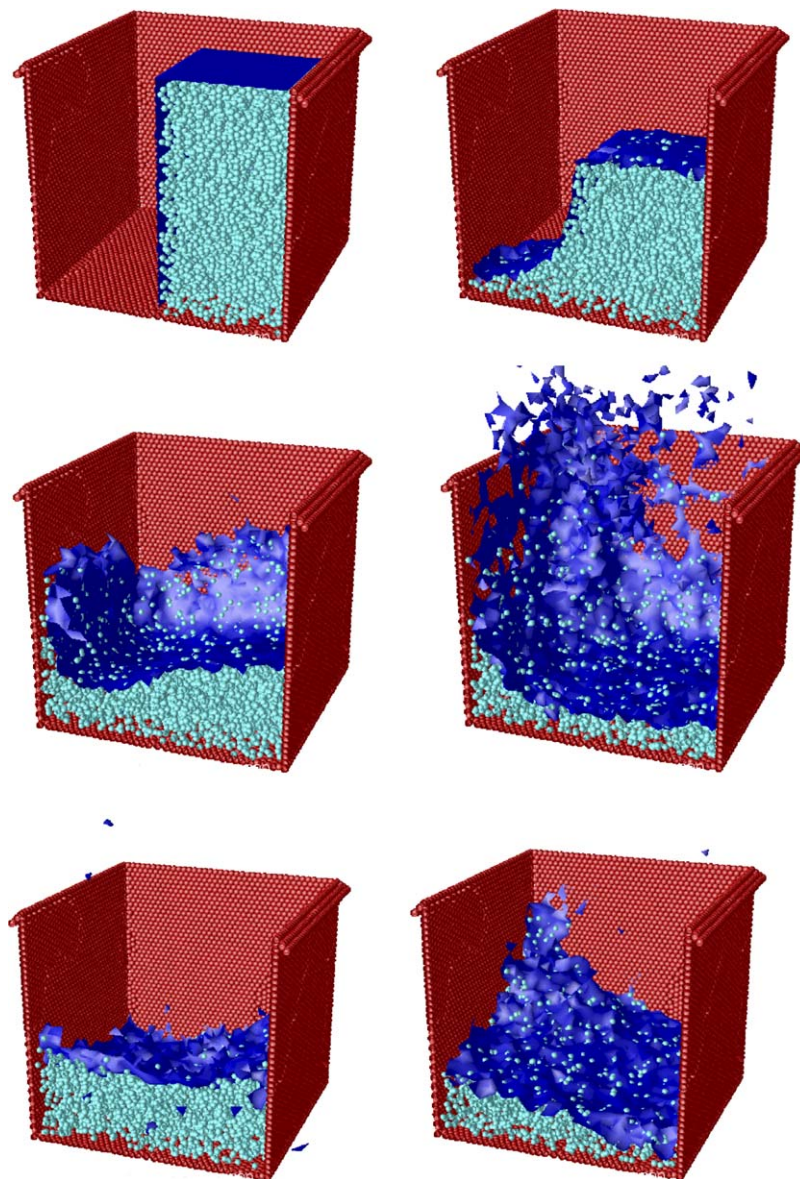


Fig. 11. Water column collapse.

In Fig. 10 the computing time for the EDT connectivity and the alpha-shape boundary recognition is shown for different structured and non-structured cubic array. It is clearly shown that the EDT and the alpha-shape method solve both problems in order $n^{1.1}$.

In particular, in a standard PC of 1 GHz, the computing time in seconds for the mesh generation and boundary recognition is

$$t \text{ (s)} = Cn^{1.1} = 0.00283n^{1.1}.$$

This time must be compared with the computing time need to solve a four nonlinear system of n equations with n unknowns. Obviously, this is problem dependent. However for standard 3D problems using a conjugate gradient iterative method, an optimist number of operations to achieve a reasonable convergence error is of the order $Cn^{1.6}$. Considering three iterations per time step in a standard PC of 1 GHz, realistic value for C is 10^{-6} . Then

$$t \text{ (s)} = 10^{-6} \times 3 \times 4n^{1.6}.$$

Table 3 shows a comparison of the computing time for different number of nodes for a standard CFD formulation using a fractional step method and for the PFEM. In the column listed as other methods without mesh update, any Eulerian CFD formulation with any fully implicit fractional-step method is included. In the PFEM column, any mesh or meshless method with an evaluation of the connectivity order n should be equivalent. It is clear that for large problems ($n \geq 10^6$), the computing time needed to evaluate a new mesh at each time step is not important compared with that required to solve the nonlinear system.

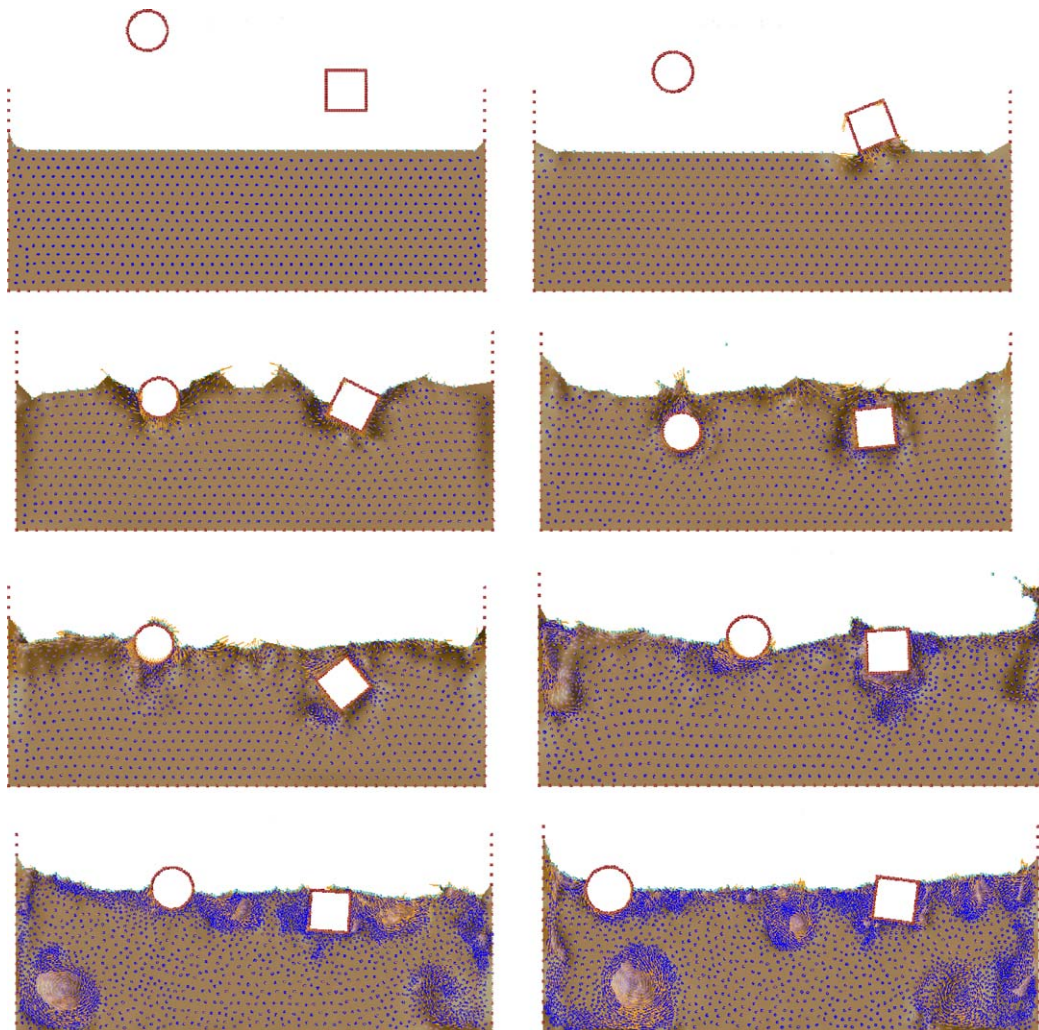


Fig. 12. Two rigid bodies falling down into a recipient with water.

7. Some PFEM examples showing the need for fast nodal connectivity evaluation

Fig. 11 shows a 3D water column in a square container. When the imaginary removable boards containing the water slid up, the water collapse starts. Viscosity has been neglected in this example. The container is represented in the figure by red spheres and the fluid by sky blue ones. The free surface detected by the alpha-shape method is represented by a blue surface. During the different time steps, fluid separation and breaking waves may be observed.

The problem was solved using the PFEM described above.

The example shows a typical problem in which the variation of the domain is so important that the use of a method that permanently updates the shape functions is fully justified.

Fig. 12 shows two rigid bodies falling down in a recipient with water. The bodies are subjected to pressure and viscous forces acting on their boundaries. The resultant of the fluid forces and the weight of the cube are applied to the centre of the cube and the cylinder. These forces govern the displacement of the bodies which is computed by solving the dynamic equations of motion as described in the fractional step algorithm of Section 5. Here the moving cube and the cylinder contours define a boundary condition for the fluid particles at each time step.

Initially the solids fall down freely due to the gravity forces. Once in contact with the water surface, the alpha-shape method recognizes the different boundary contours, which are shown with a thick line in the figure. The pressure and the viscous forces are evaluated in the entire domain and in particular on the cube and cylinder contours. The fluid forces introduce a negative acceleration in the vertical motion until, once the bodies are completely inside the water, the vertical velocity becomes zero. Then, Archimides forces bring the cylinder up to the free-surface, while the cube remains inside the fluid due to differences in the density of both bodies. Note that the method reproduces very well the interaction of the rigid bodies with the free surface as well as the overall sinking process.

We recall that in all the problems here described the mesh in the fluid domain is regenerated at each time step combining polyhedral elements as described in Section 5. Note that some fluid particles separate from the fluid domain. These particles are treated as free boundary points with zero pressure and hence fall down due to gravity.

Further examples of applications of the PFEM can be found in [11–13] and in www.cimne.com/pfem.

8. Conclusions

Particle methods combined with the shape functions of the meshless finite element method, in which the nodal connectivity is generated in a time which is linear with the number of particles, are an excellent tool to solve fluid mechanics problems, especially fluid–structure interaction situations with moving free-surfaces. The MFEM seems to be the best adapted FEM to this kind of combination. In fact the MFEM has the advantages of a meshless method concerning the easy introduction of the nodal connectivities in a bounded time of order n . The method also preserves the classical advantages of the FEM such as: (a) the simplicity of the shape functions, (b) C^0 continuity between elements, (c) an easy introduction of the boundary conditions, and (d) symmetric matrices.

Both, particle methods and the MFEM, are the key ingredients to the PFEM, which has been shown to be a very suitable method to solve fluid–structure interaction problems including free-surface, breaking waves, flow separations, contact problems and collapse situations.

The authors have used the PFEM as an example to demonstrate the question introduced in the title of this paper. The answer to the main questions concerning if the right question before solving a numerical problem is “To mesh or not to mesh...” is: No, in most of the applications, that is not the right question!

When the problem to be solved needs a permanently update of the nodal connectivity, the most important requirement is to ask for an algorithm giving a linear relation between the number of nodes and the number of operations to generate the connectivity. If the algorithm is based on a mesh or a meshless method is, in general, irrelevant. Both mesh based methods or meshless methods may give the correct answer provided the algorithm used for the evaluation of the node connectivity is bounded in computing time and is linear with the number of nodes.

Some particular problems exist in which the specific feature of meshless methods such as C^∞ interpolations or non-local interpolations becomes an advantage. However, these are not today the main applications of meshless methods. Indeed these topics deserve further investigation.

Unfortunately, many of the papers found in the literature regarding meshless methods are not very useful for practical purposes. Some times because they solve problems in which the mesh generation is not an important issue in the problems presented, or simply because they use an algorithm to generate the connectivity between nodes that has not a linear relation between the number of operations and the number of nodes involved. In this case, the generation of a mesh would be more efficient.

Quite frequently, papers do not give any information concerning the algorithm used to generate the nodal connectivity, or do not justify properly the need of some special interpolation for a particular application. As explained in this paper, this

information is essential in order to satisfy the “raison d’être” of a meshless method, as well as to assess the potential for practical purposes.

Acknowledgements

Thanks are given to Facundo Del Pin, Nestor Calvo, Romain Aubry and Miguel Angel Celigueta for computing the numerical results with the PFEM presented.

References

- [1] J.H. Argyris, H. Kelsey, H. Kamel, Matrix methods of structural analysis: a précis of recent developments, in: B. Fraeijs de Veubeke (Ed.), *Matrix Methods of Structural Analysis*, AGARDograph 72, Pergamon Press, Oxford, 1964, pp. 1–164.
- [2] B. Fraeijs de Veubeke, Displacement and equilibrium models in the finite element method, in: O.C. Zienkiewicz, G.S. Holister (Eds.), *Stress Analysis*, John Wiley & Sons, London, 1965, pp. 145–197 (Chapter 9).
- [3] O.C. Zienkiewicz, Y.K. Cheung, *The Finite Element Method in Continuum and Structural Mechanics*, McGraw Hill, New York, 1967, 272 pp.
- [4] B. Nayroles, G. Touzot, P. Villon, Generalizing the FEM: Diffuse approximation and diffuse elements, *Comput. Mech.* 10 (1992) 307–318.
- [5] T. Belytschko, Y. Liu, L. Gu, Element free-Galerkin methods, *Int. J. Numer. Methods Engrg.* (1994) 37229–37256.
- [6] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, C. Sacco, A stabilized finite point method for analysis of fluid mechanics problems, *Comput. Methods Appl. Mech. Engrg.* 39 (1996) 315–346.
- [7] S.R. Idelsohn, M.A. Storti, E. Oñate, Lagrangian formulations to solve free surface incompressible inviscid fluid flows, *Comput. Methods Appl. Mech. Engrg.* 191 (2001) 583–593.
- [8] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, A finite point method in computational mechanics. Applications to convective transport and fluid flow, *Int. J. Numer. Methods Engrg.* 39 (22) (1996) 3839–3886.
- [9] S.R. Idelsohn, E. Oñate, N. Calvo, F. Del Pin, The meshless finite element method, *Int. J. Numer. Methods Engrg.* 58 (4) (2003).
- [10] N. Calvo, S.R. Idelsohn, E. Oñate, Polyhedrization of an arbitrary 3D point set, *Comput. Method Appl. Mech. Engrg.* 192 (2003) 2649–2667.
- [11] S.R. Idelsohn, E. Oñate, F. Del Pin, The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves, *Int. J. Numer. Methods Engrg.* 61 (7) (2004) 964–989.
- [12] S.R. Idelsohn, E. Oñate, F. Del Pin, A Lagrangian meshless finite element method applied to fluid–structure interaction problems, *Comput. Struct.* 81 (2003) 655–671.
- [13] E. Oñate, S.R. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method: an overview, *Int. J. Comput. Methods* 1 (2) (2004) 267–307.
- [14] J. Orkisz, Recent advances in the meshless finite difference method, in: *Proceedings of the 2nd ECCM, Cracow, 26–29 June 2001*.
- [15] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics, theory and application to non-spherical stars, *Mon. Nat. Roy. Astr. Soc.* 181 (1997) 375–389.
- [16] W.K. Liu, J.S. Li, J.S. Adee, T. Belytschko, Reproducing kernel particle methods for structural dynamics, *Int. J. Numer. Methods Engrg.* 38 (1995) 1655–1679.
- [17] J.M. Melenk, I. Babuiska, The partition of unity element method: basic theory and Applications, *Comput. Methods Appl. Mech. Engrg.* 139 (1996) 289–314.
- [18] S. De, K.J. Bathe, Towards an efficient meshless computational technique: the method of finite spheres, *Engrg. Comput.* 18 (2001) 170–192.
- [19] N. Sukumar, B. Moran, T. Belytschko, The natural element method in solid mechanics, *Int. J. Numer. Methods Engrg.* 43 (1998) 839–887.
- [20] C. Duarte, J.D. Oden, Hp clouds—a meshless method to solve boundary value problems, *Tech. Rep. 95-05*, Texas Institute for Computational and Applied Mechanics, University of Texas, 1995.
- [21] H. Edelsbrunner, E.P. Mücke, Three-dimensional alpha-shape, *ACM Trans. Graph.* 3 (1994) 43–72.
- [22] V. Belikov, A. Semenov, Non-Sibsonian interpolation on arbitrary system of points in Euclidean space and adaptive generating isolines algorithm, in: *Numerical Grid Generation in Computational Field Simulation*, Proc. of the 6th Intl. Conf., Greenwich University, July 1998.
- [23] R. Codina, Pressure Stability in fractional step finite element methods for incompressible flows, *J. Comput. Phys.* 170 (2001) 112–140.
- [24] E. Oñate, A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation, *Comput. Methods Appl. Mech. Engrg.* 182 (1–2) (2000) 355–370.
- [25] W.K. Liu, W. Han, H. Lu, S. Li, J. Cao, Reproducing kernel element method. Part I: Theoretical formulation, *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 989–1011.
- [26] D. Qian, W.K. Liu, A multiscale meshfree method for the mechanical analysis of low dimensional nanostructures, in: *6th International Conference on Modeling and Simulation of Microsystems, MSM 2003*, Computational Publication, vol. 2, Cambridge, MA, USA, 2003, pp. 550–553.